Let

- f_θ be a language model.
- T be the tokenizer associated with f_θ.
- $T(w)$ denote the sequence of token IDs of a word w.
- $T(s)$ denote the sequence of token IDs of a sentence s.
- $\log \operatorname{softmax}(\mathbf{z})$ denote the vector of log-probabilities over the vocabulary obtained from logits $\mathbf{z}$.
- $\operatorname{LogSumExp}(a_1, \ldots, a_K)$ denote $\log\Bigl(\sum_{k=1}^{K} e^{a_k}\Bigr)$.

Algorithm 1 , dynamic case

Require: sentence s, model f_θ, tokenizer T, set of words $\mathcal{W}$
Ensure: mapping Result : $\mathcal{W} \to \mathbb{R}$ of average log-probabilities

$x \leftarrow T(s)$ ▷ $x = (x_1, \ldots, x_n)$, sentence token IDs
$n \leftarrow |x|$ ▷ number of tokens in the sentence
Result $\leftarrow \emptyset$
for each word $w \in \mathcal{W}$ **do**
  $\tau(w) \leftarrow T(w)$ ▷ $\tau(w) = (t_1, \ldots, t_L)$, token IDs of w
  if $|\tau(w)| = 0$ **then**
    Result$[w] \leftarrow$ NaN
    continue to next w
  $\mathcal{S} \leftarrow \emptyset$ ▷ list of joint log-probabilities for each insertion position
  $K \leftarrow n + 1$ ▷ number of insertion positions
  for $k = 0$ to n **do**
    prefix $\leftarrow (x_1, \ldots, x_k)$
    context $\leftarrow$ prefix
    $\ell_{\text{joint}} \leftarrow 0$ ▷ joint log-probability for this insertion position
    for $j = 1$ to L **do**
      if $|\text{context}| = 0$ **then**
        eval_context $\leftarrow$ (BOS) ▷ BOS token ID from tokenizer
      else
        eval_context $\leftarrow$ context
      $\mathbf{z} \leftarrow f_\theta(\text{eval\_context})$ ▷ logits for next token
      $\ell \leftarrow \log \text{softmax}(\mathbf{z}_{\text{last}})$
      $\ell_{\text{token}} \leftarrow \ell[t_j]$
      $\ell_{\text{joint}} \leftarrow \ell_{\text{joint}} + \ell_{\text{token}}$
      context $\leftarrow$ context $\|\ t_j$ ▷ append t_j to context
    append ℓ_{joint} to $\mathcal{S}$
  if $|\mathcal{S}| > 0$ **then**
    $\ell_{\text{sum}} \leftarrow \text{LogSumExp}(\mathcal{S})$
    $\ell_{\text{avg}} \leftarrow \ell_{\text{sum}} - \log K$
    Result$[w] \leftarrow \ell_{\text{avg}}$
  else
    Result$[w] \leftarrow$ NaN
return Result

Algorithm 2 , static case

Require: sentence s, model f_θ, tokenizer T, set of words $\mathcal{W}$

Ensure: mapping Result : $\mathcal{W} \to \mathbb{R}$ of average log-probabilities

$x \leftarrow T(s)$ ▷ $x = (x_1, \ldots, x_n)$, sentence token IDs

$n \leftarrow |x|$

$\mathbf{Z} \leftarrow f_\theta(x)$ ▷ $\mathbf{Z} \in \mathbb{R}^{n \times |\mathcal{V}|}$, logits over vocabulary $\mathcal{V}$

$\mathbf{L} \leftarrow \log \mathrm{softmax}(\mathbf{Z})$ along the vocabulary dimension ▷ $\mathbf{L} \in \mathbb{R}^{n \times |\mathcal{V}|}$ is the static log-probability table

Result $\leftarrow \emptyset$

for each word $w \in \mathcal{W}$ **do**

$\tau(w) \leftarrow T(w)$ ▷ $\tau(w) = (t_1, \ldots, t_L)$, token IDs of w

$L_w \leftarrow |\tau(w)|$

if $L_w = 0$ **then**

Result$[w] \leftarrow$ NaN

continue to next w

$\mathcal{S} \leftarrow \emptyset$ ▷ list of pseudo-joint log-probabilities

$K \leftarrow n - L_w + 1$ ▷ number of sliding-window positions

for $k = 1$ to K **do**

$\ell_{\text{joint}} \leftarrow 0$

for $j = 1$ to L_w **do**

$i \leftarrow k + j - 1$ ▷ sentence position

$\ell_{\text{token}} \leftarrow \mathbf{L}_{i,\, t_j}$ ▷ log-probability of t_j at position i

$\ell_{\text{joint}} \leftarrow \ell_{\text{joint}} + \ell_{\text{token}}$

append ℓ_{joint} to $\mathcal{S}$

if $|\mathcal{S}| > 0$ **then**

$\ell_{\text{sum}} \leftarrow \mathrm{LogSumExp}(\mathcal{S})$

$\ell_{\text{avg}} \leftarrow \ell_{\text{sum}} - \log K$

Result$[w] \leftarrow \ell_{\text{avg}}$

else

Result$[w] \leftarrow$ NaN

return Result